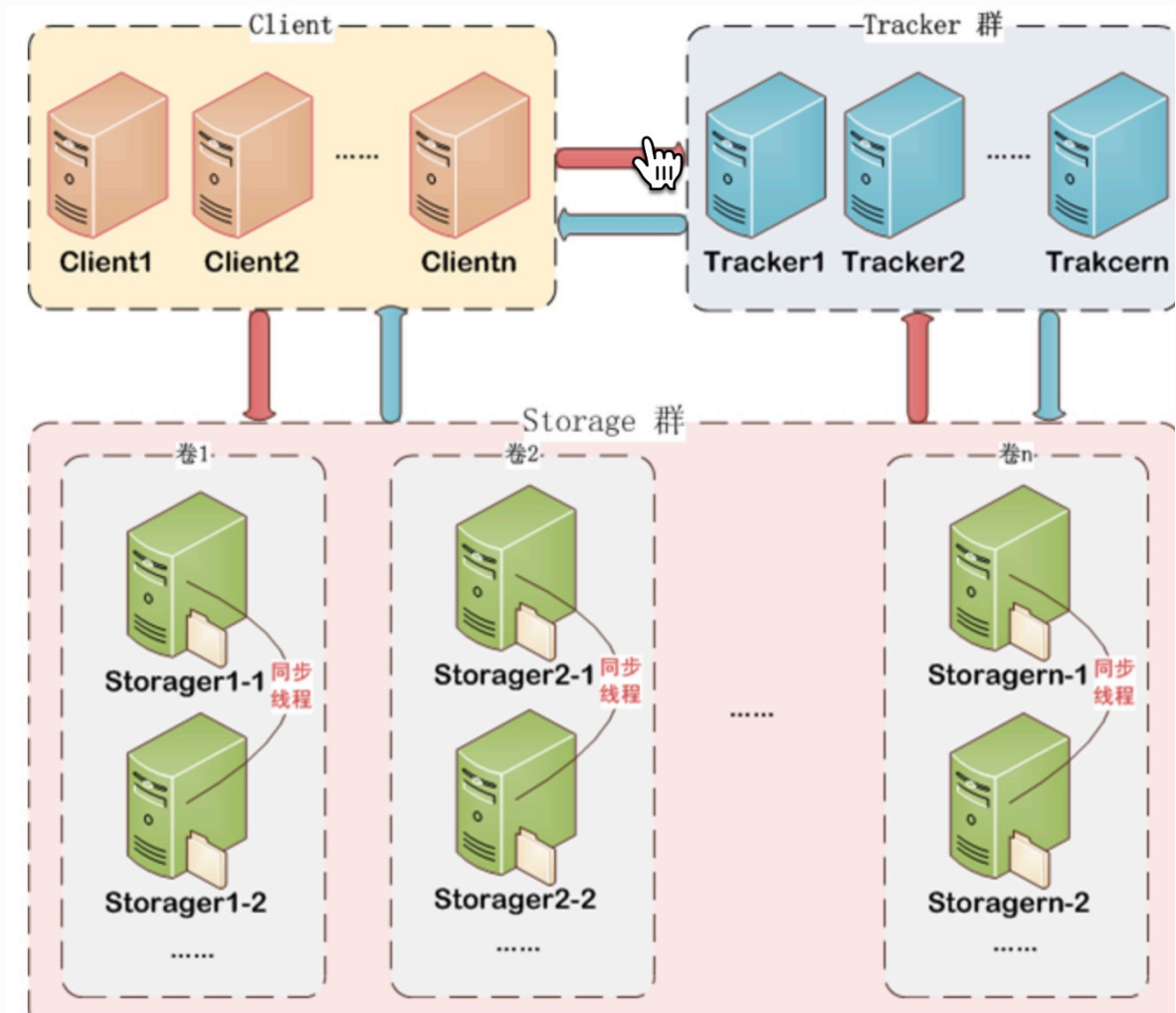


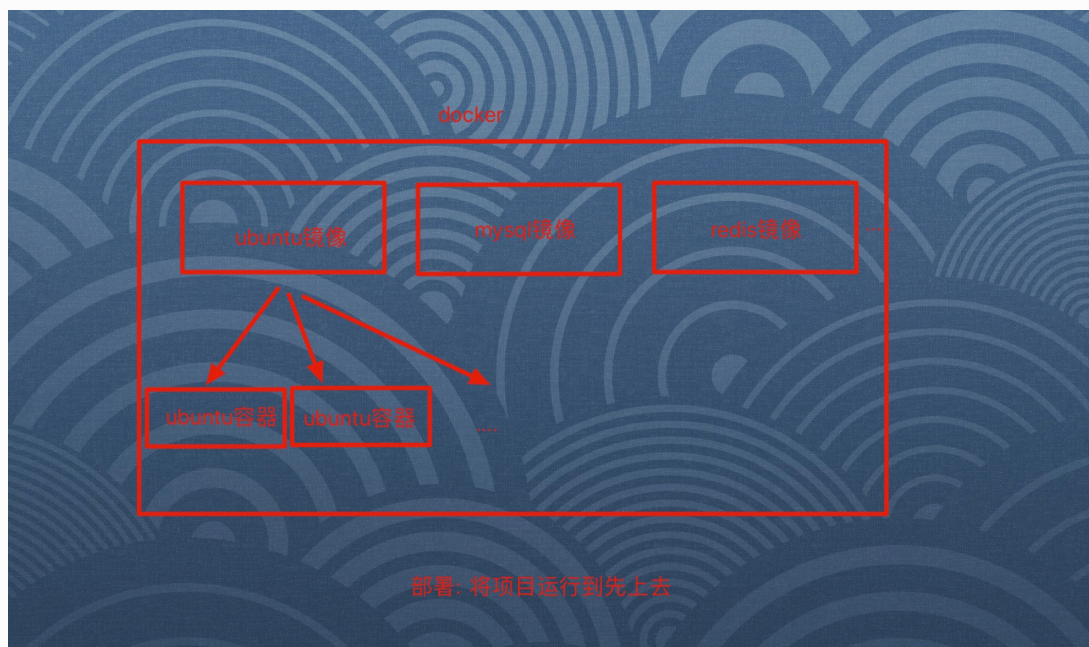
1, FastDFS 介绍

- 目的: 知道FASTDFS作用,以及组成
- 作用: 分布式的文件存储系统
- 作用: Tracker ,Storage



2, Docker 介绍

- 目的: 理解docker的作用和组成
 - 作用: 容器化技术
 - 组成: 镜像 + 容器
 - 特点:
 - 入门简单
 - 开发效率高
 - 可移植性强



3,Docker安装(见文档)

- 目的: 能够参考文档,安装Docker

4,Docker镜像(见文档)

- 目的: 能够对镜像进行增删改查

5,Docker容器(见文档)

- 目的: 能够对镜像进行增删改查

6,FastDFS安装

- 目的: 参考文档,安装storage,tracker容器和镜像
- 注意点:
 - tracker: 运行的端口是22122
 - storage: 运行的端口是8888

7,图片存储测试

- 目的: 能够在终端中进行图片上传的测试
- 操作流程:
 - 1, 安装fdfs的客户端

```
1 $ pip install fdfs_client-py-master.zip
2 $ pip install mutagen
3 $ pip isntall requests
```

- 2,创建fdfs包,拷贝client.conf配置文件(utils/fdfs/client.conf)
- 3,在终端测试即可

```
1
2 In [1]: from fdfs_client.client import
        Fdfs_client

3
4 In [2]: #1,创建对象

5
6 In [3]: client =
        Fdfs_client('/Users/heJing/Desktop/classes2/day07/meiduo12/meiduo_mall/meiduo_mall/utils/fdfs/client.conf')

7
8 In [4]: #2,上传

9
10 In [5]: result =
        client.upload_by_filename('/Users/heJing/Desktop/上课资料2/day07/3_资料/07_图片素材/hello.jpeg')

11 getting connection
12 <fdfs_client.connection.Connection object at 0x10f530b00>
13 <fdfs_client.fdfs_protocol.Tracker_header object at 0x10f5309e8>
14
```

```
15 In [6]: result

16 Out[6]:
17 {'Group name': 'group1',
18  'Remote file_id':
19  'group1/M00/00/02/rBAMhlzvmOCAA6hJAACdrkQkokk2
20  6.jpeg',
21  'Status': 'Upload succeeded.',
22  'Local file name': '/Users/heJing/Desktop/上课资料2/day07/3_资料/07_图片素材/hello.jpeg',
23  'Uploaded size': '39.00KB',
24  'Storage IP': '172.16.12.134'}

24 In [7]:
```

8.测试数据添加

- 目的: 能够将测试数据添加到数据库中

9.首页商品分类

- 目的: 能够理解首页分类数据的查询和展示即可
- 操作流程:
- 1, 编写类视图组装数据(contenxt/views.py)

```
1 class IndexView(View):
2     def get(self, request):
3
4         #1, 定义字典
5         categories = {}
6
7         #2, 查询所有的频道组
8         channels =
9         GoodsChannel.objects.order_by('group_id', 'sequence')
10
11         #3, 遍历频道组, 组装数据
```

```

11         for channel in channels:
12
13             #3.1 取出组的编号
14             group_id = channel.group_id
15
16             #3.2 组装好一个分类的字典
17             if group_id not in categories:
18                 categories[group_id] =
19                 {"channels": [], "sub_cats": []}
20
21             #3.3 添加一级分类到channels
22             catetory = channel.category
23             catetory_dict = {
24                 "id": catetory.id,
25                 "name": catetory.name,
26                 "url": channel.url
27             }
28             categories[group_id]
29             ["channels"].append(catetory_dict)
30
31             #3.4 添加二级分类三级分类
32             for cat2 in catetory.subs.all():
33                 categories[group_id]
34                 ["sub_cats"].append(cat2)
35
36             context = {
37                 "categories": categories,
38                 # "contents": "?"
39             }
40
41             return
42
43     render(request, 'index.html', context=context)

```

10. 首页商品广告

- 目的: 能够拼接数据, 渲染前端页面
- 操作流程:
 - 1, 编写类视图代码(contens/views.py)

```

1 class IndexView(View):

```

```

2         def get(self, request):
3             ...
4
5             #4, 拼接广告数据
6             contents = {}
7             content_categories =
ContentCategory.objects.all()
8             for content_category in
content_categories:
9                 contents[content_category.key] =
content_category.content_set.order_by('sequence')
10
11             #5, 拼接数据, 返回响应
12             context = {
13                 "categories": categories,
14                 "contents": contents
15             }
16
17             return
render(request, 'index.html', context=context)

```

- 2, 替换index.html模板文件即可